Paper

Automatic tactical network node configuration with XML and SNMP

Marek Małowidzki and Przemysław Bereziński

Abstract— In the paper, we describe a "plug-and-play" configuration of nodes of a tactical network on the basis of XML configuration templates and a network plan, developed during the network planning process. We present the concept of a configuration repository, an XML-based database that stores network structure and configuration data, and describe how the Simple Network Management Protocol is used to apply the settings to network devices. We also comment on a possible use of the next-generation NETCONF protocol for such a task.

Keywords— network management, network configuration, tactical network, SNMP, NETCONF.

1. Introduction

The typical operation of a tactical network usually consists of two main phases. The first, *planning* phase, requires defining all the important details about the network's configuration: where the nodes are placed, how they are connected, how the underlying networking technologies are configured. The second phase, the *operation* one, assumes the network is up and running. Between these two phases the developed plan should be appropriately mapped onto networking devices and their parameters. This is what the paper is about – it describes our approach to a transparent, "plug-and-play" type of network nodes configuration.

The paper is organized as follows. First, we describe the tactical network. Then, we give an overview of the most important components that support our approach, namely, the configuration repository and node configuration templates. Later in the paper, we discuss the network planning and configuration processes. We then comment on how we could benefit from using the network configuration (NETCONF) protocol. Finally, we discuss future work and end the paper with conclusions.

2. The tactical network

The general outline of the network's architecture is presented in Fig. 1. The network core is built using asynchronous transfer mode (ATM) technology, which integrates IP traffic generated by a management system and computer networks (local area networks – LANs), and telephone traffic coming from integrated services digital network (ISDN) subscribers. Besides, the ATM core provides some military-specific quality of service (QoS) features and improves fault tolerance.

Network nodes are not fully mobile, as they do not work during motion. Only radio access points (RAPs) and radio users (RU) are fully mobile and can communicate while in motion.

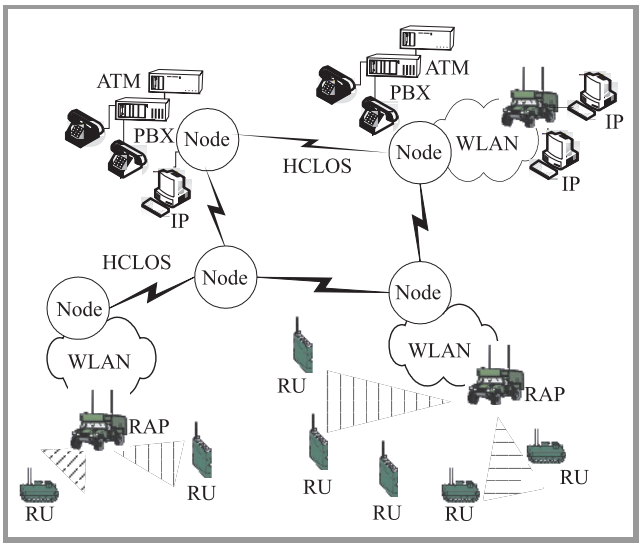

Fig. 1. The tactical network.

In this work, we describe the automatic configuration of the network core, which requires handling the above-mentioned technologies and their relationships to make the network function properly. The correct configuration requires a number of various network devices and services to be configured.

3. The configuration repository

The configuration repository [1] is a flexible database that contains information about the tactical network's structure and current configuration. The structure is defined in an extensible markup language (XML) template. The template describes network objects (link, interface, device, vehicle, and node types) and their relationships (containment and connections, e.g., what kind of equipment a given vehicle contains and how the devices are connected). The repository also contains the current network configuration. Additionally, the repository provides a number of services for client applications: MT-safety, state propagation, privileged

and non-privileged interface, and others. (For more information, refer to [1].) Note that the repository is independent of the network type and may be adapted, through providing custom templates, for other network types, and even to other scenarios. In fact, we have used the repository successfully in a number of other projects – a recent example was the implementation of the SecureSOA demonstrator during the Coalition Warrior Interoperability Demonstration 2006 (CWID'2006) in Lillehammer, when the repository was employed to model tactical situation [2].
The configuration repository is implemented as a Microsoft .NET 2.0 component and extensively uses XML technology. The same component is used during network planning and later, during the management phase.

4. Node configuration templates

The tactical network employs a number of advanced technologies, and there are a large number of possible configuration settings for every network device. Fortunately, the number of typical options for inter-node connections is limited – speaking in other words, only a subset of possible options may be applied. Thus, it was possible to follow the approach the configuration repository is based upon and define XML *node configuration templates* for the network. The templates are used during both network planning and configuration and they contain a number of XML elements that automate both processes. They should be prepared in advance by network management engineers with deep knowledge of the network equipment.

5. Network planning

Network planning is supported by a dedicated network planning application (Fig. 2). The application uses the configuration repository to learn the network structure, node, vehicle and device types, possible connection options, etc. The application is independent of details, e.g., when the structure changes, or a new equipment is introduced, usually only the configuration repository (and also, the node configuration templates) must be modified. The application supports placing nodes on a digital map, configuring their parameters and links, configuring the network (IP, ATM addressing; ISDN numbering) and also implements some advanced features that support radio communication for high capacity line-of-sight (HCLOS) radio lines and WLANs (frequency assignment, terrain profiles, etc.).
For each network node type, the configuration repository lists its *node external interfaces* – if there are any – that is, the interfaces of the node's devices that are used to connect to other nodes. The node configuration templates, on the other hand, contain a wizard-type sequence of XML elements that guide through the process of defining the inter-node connection. These elements, generally, are a sequence of – usually nested – choices that a network planner makes to configure the connection, selecting appropriate values for

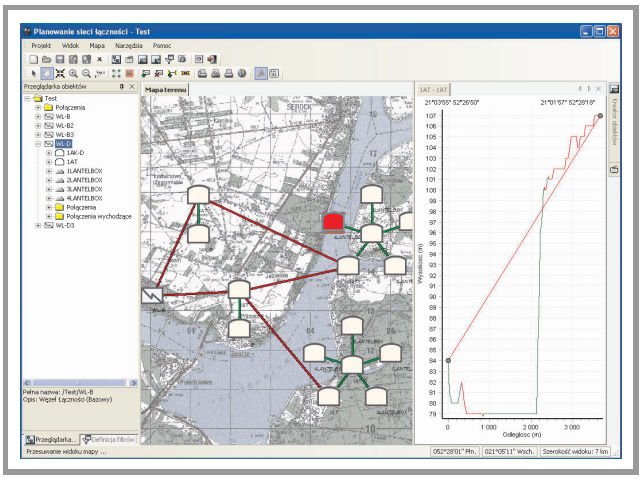

Fig. 2. Network planning application.

crucial parameters and going through subsequent choices to complete the configuration process. For example, configuring HCLOS link between two nodes requires defining radio line parameters (frequencies, link capacity, modulation) and ATM settings (IMA group settings, signalization type, signalization side, clock, etc.).

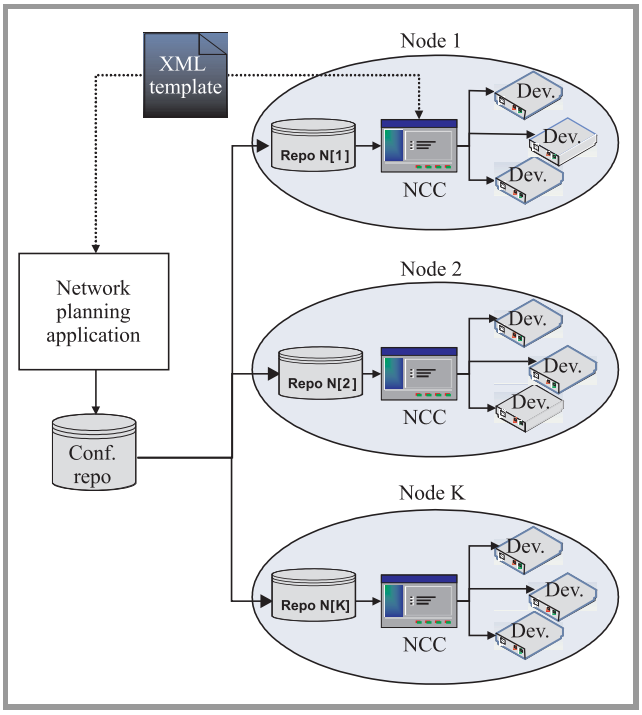

Fig. 3. Use of the XML node configuration templates during network planning and configuration.

After the planning process has finished, the network configuration is stored in a binary file. The file is delivered to a network management application, which manages the whole network. Additionally, for each node, a binary *node configuration file* is created. This file is used by the node staff to prepare the node to work. Later, the same file is used by a node management application. The binary files

are generally equivalent to the content of the configuration repository (for example, a simple .NET *serialization* may be used here). The whole process is depicted in Fig. 3.

6. Network node configuration

After the node's vehicles have been prepared to work in their destination positions, the staff in every vehicle uses a wizard-type application called node configuration creator (NCC) that reads the binary node configuration file and configures network equipment within each vehicle (this is the initial node configuration phase; however, the same process may be applied later, to re-configure the node if necessary). First, static IP addresses are set through a serial RS-232C connections. Then, the Simple Network Management Protocol (SNMP) [3] is used to configure the devices. In most cases, the human operator only needs to observe the configuration process and press an "OK" button a number of times; sometimes, however, the application asks the operator to perform some additional tasks manually (e.g., when a device of an older type needs to be reset after changes have been applied). Of course, as vehicles are not fully automated, additional manual actions are necessary, e.g., to adjust antennas to appropriate azimuths.

The NCC uses the same XML configuration template file that has been used during network planning. The template contains a dedicated entry for every parameter of a given device type; the entry provides SNMP-related data required by the application. Figure 4 shows a fragment of the template file that illustrates the idea.

```
<Config>
  <NetworkElement>atmEquipment.AXD</NetworkElement>
  <InterfaceType>interface.e1.1E1</InterfaceType>
  <Parameter>
    <RepoName>phyIfSignalingSide</RepoName>
    <SnmpOID>1.3.6.1.4.1.415.2.3.2.1.4.</SnmpOID>   <!-- object identifier -->
    <SnmpType>SnmpInt32</SnmpType>                  <!-- variable type      -->
    <Choice>
      <RepoValue>Network</RepoValue>
      <RepoRemoteValue>User</RepoRemoteValue>
      <SnmpValue>2</SnmpValue>                      <!-- variable value     -->
    </Choice>
    <Choice>
      <RepoValue>User</RepoValue>
      <RepoRemoteValue>Network</RepoRemoteValue>
      <SnmpValue>3</SnmpValue>                      <!-- variable value     -->
    </Choice>
```

Fig. 4. XML configuration entries related to SNMP.

The NCC is a Microsoft .NET Framework 2.0 application that employs the SNMP++.NET [4] open-source SNMP component. The component is itself based on the SNMP++ [5] library. SNMPv3 (SNMP version 3, supporting authentication and encryption) is the default version and is used for all devices that support SNMPv3.

The advantage of this approach is that a single, flexible application is able to configure all device types in a uniform way. In fact, the NCC does not need to differentiate the device types, as the configuration algorithm for every device is the same. Thus, adding another device type to a network merely requires the configuration template file to be appropriately extended. This is, generally, the same idea that the configuration repository uses – shifting as much burden as possible to a template and avoiding changes in software, even after the network equipment or structure changes.

Unfortunately, there are also some drawbacks. The drawbacks generally stem from the fact how the SNMP protocol operates. Most operations involve SNMP tables and their indexes; also, various parameters (mapped to SNMP variables) are often interrelated and the order in which they are set is significant. This causes the template to be more complex and the person who creates the template must be aware of all these details. Yet another problem are the limited error reporting capabilities of the SNMP protocol – if anything goes wrong, it is generally impossible for the application to display an informative message about the problem cause (in one of the SNMP agents we implemented additional, non-standard error reporting functions, but they are unavailable in most devices).

As it was mentioned above, the planning and configuration phases are limited to configuring connections between nodes and, to some degree, connections between nodes' vehicles. It is assumed that internal settings within vehicles (e.g., the way their internal devices cooperate) is fixed with no need for any changes.

7. The NETCONF

According to [6], the Network Configuration Protocol [7] provides a means to install, manipulate or delete configuration of network devices. It uses XML for data encoding and a simple RPC-type request-response model, which may be implemented atop any transport layer that meets some criteria. NETCONF is likely to become the standard, next-generation network management protocol and replace SNMP in some foreseeable future.

Application of NETCONF in our network would yield a number of benefits:

- **XML technology**. With XML and extensible stylesheet language transformations (XSLT), it would be possible to prepare configurations for devices in advance, during the planning phase. Note that usually, configurations for the same device type only differ in a number of details.
- **Configuration management**. NETCONF provides a means to define multiple configurations for a device and then easily switch between them (e.g., return to a previous one in case of error or pre-planned change). Additionally, a complete configuration may be loaded onto and read from a device in a single step. On the other hand, there is no a similar notion of "configuration" in SNMP.
- **Protocol operation**. NETCONF allows a configuration to be loaded to and read from a device in a single step. Additionally, NETCONF has better mechanisms that help synchronizing multiple managers (operating on the same device) and its error reporting capabilities are far better than in SNMP.

We believe that NETCONF would further simplify the whole process. Unfortunately, it is not supported in network devices. One of the possible reasons could be the fact that the devices often employ embedded systems, with limited memory and processing capabilities.

8. Future work

Every new network service requires this work to be extended. User mobility, i.e., the ability to migrate between various locations and terminal types, requires a number of services to be properly planned and configured (LDAP directory servers, H.323 gatekeepers, ISDN-IP and IP-radio gateways). This still remains to be done.

Additionally, we consider adding capabilities to configure internal interfaces. This would enable to, for example, initially configure a freshly produced vehicle. Currently, this "factory", default configuration must be applied using other means.

Yet another issue is the ability to rollback changes in case of error. This is not implemented. In fact, the only way is just to configure a node (or, vehicle) again using a previous file.

Finally, the configuration process is currently one-way: it is possible to "inject" a configuration from the configuration repository into devices but there is no capability to read it back (i.e., fill the repository with the current configuration read from devices).

9. Conclusions

In our tactical network, there are a number of typical network configuration scenarios. Our goal was to support these scenarios in such a way to automate the network planning and network configuration processes as far as possible. We have succeeded to ease the planning phase – although still some work is required here – and to significantly simplify the configuration phase. The node configuration creator enables easy and quick node configuration and reconfiguration. The node staff does not need to be highly skilled in modern networking technologies and, additionally, the possibility of introducing human errors is greatly reduced. As it was mentioned above, additional work is undergoing to extend the planning and configuration process to include critical node server applications.

The strength of our approach is that it is technology-independent and could be used for planning and configuration of other, even significantly different, network types of similar complexity. This of course also means that changes to the current network (e.g., new equipment, new vehicle or node types, etc.) could be easily addressed.

Our approach assumes that the SNMP protocol is used for actual configuration of network devices. In fact, we believe that the approach perfectly complies with the ideas the NETCONF protocol is based on. Unfortunately, due to lack of support for NETCONF in network equipment, employing such a combination is currently impossible.

References

[1] M. Małowidzki, "XML-based configuration repository for a mobile broadband network", in *Proc. World Multi-Conf. Syst. Cyber. Inform. SCI'2004*, Orlando, USA, 2004.

[2] M. Małowidzki, K. Liponoga, P. Sobonski, R. Goniacz, J. Sliwa, R. Piotrowski, and M. Amanowicz, "Secure information sharing in a tactical network", in *Proc. Milit. CIS Conf. MCC'06*, Gdynia, Poland, 2006.

[3] D. Harrington, R. Presuhn, and B. Wijnen, "An architecture for describing simple network management protocol (SNMP) management frameworks", RFC-3411, Dec. 2002.

[4] SNMP++.NET, http://maom_onet.republika.pl/snmp/snmp_ppnet/

[5] SNMP++, http://www.agentpp.com/snmp_pp3_x/snmp_pp3_x.html

[6] R. Enns, "NETCONF configuration protocol", RFC-4741, Dec. 2006.

[7] IETF network configuration (NETCONF) group, http://www.ietf.org/html.charters/netconf-charter.html

Marek Małowidzki obtained M.Sc. degree in 1996 from the Faculty of Electronics of the Warsaw University of Technology (WUT). Currently employed in Military Communication Institute, Zegrze, Poland. His main interests include object-oriented and component software technologies, network management and the Internet.

e-mail: m.malowidzki@wil.waw.pl
Military Communication Institute
05-130 Zegrze, Poland

Przemysław Bereziński received M.Sc. degree in 2006 from the Faculty of Physics, Mathematics and Information Technology of Łódź University of Technology. At the moment he is a researcher in Military Communication Institute, Zegrze, Poland. His main areas of interest include object-oriented technologies, databases, network management and IP-based networks.

e-mail: p.berezinski@wil.waw.pl
Military Communication Institute
05-130 Zegrze, Poland